frontiers | Frontiers in Blockchain

TYPE Original Research
PUBLISHED 05 September 2025
DOI 10.3389/fbloc.2025.1474329

OPEN ACCESS

EDITED BY
Mubashir Husain Rehmani,
Munster Technological University, Ireland

REVIEWED BY
Chin-Ling Chen,
Chaoyang University of Technology, Taiwan
Qin Wang,
Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia

*CORRESPONDENCE
Kyung-Hyune Rhee,
khrhee@pknu.ac.kr

†These authors have contributed equally to this work

RECEIVED 01 August 2024
ACCEPTED 18 August 2025
PUBLISHED 05 September 2025

CITATION
Keo R, Firdaus M and Rhee K-H (2025) A secure rubber supply chain management system based on hyperledger fabric blockchain: a use case in Cambodia.
Front. Blockchain 8:1474329.
doi: 10.3389/fbloc.2025.1474329

A secure rubber supply chain management system based on hyperledger fabric blockchain: a use case in Cambodia

Ratanak Keo[1†], Muhammad Firdaus[2,3,4†] and Kyung-Hyune Rhee[5*]

[1]Department of IT Convergence and Application Engineering, Pukyong National University, Busan, Republic of Korea, [2]Industrial Science Technology Research Center, Pukyong National University, Busan, Republic of Korea, [3]Department of Intelligence Computing, Dong-Eui University, Busan, Republic of Korea, [4]School of Computing, Telkom University, Bandung, Indonesia, [5]Division of Computer Engineering and AI, Pukyong National University, Busan, Republic of Korea

Conventional traceability systems in Cambodian agriculture, such as rubber, rice, and cassava supply chain management, often fail to provide timely data and fair market prices, leading to mistrust among stakeholders, exploitation by intermediaries, and hindering progress towards ethical sourcing and sustainable practices. This paper demonstrates the potential of integrating blockchain, IoT, the InterPlanetary File System (IPFS), and Differential Privacy (DP) techniques to empower Cambodian rubber farmers. By leveraging Hyperledger Fabric (HLF), the proposed system enhances transparency and facilitates fair pricing by forming decentralized rubber traceability from supplier to consumer. Integration with IPFS ensures secure and accessible data storage, while differential privacy techniques protect confidential data during information sharing. Our approach facilitates a model for other developing countries aiming to modernize agricultural distribution networks. Numerical results demonstrate that our system achieved the overall performance objectives, including transaction send rates, transaction throughput, and transaction latency. However, the results highlight a trade-off between privacy and performance in blockchain-based agricultural traceability systems. Specifically, while implementing differential privacy techniques enhances data confidentiality, it can slightly reduce system efficiency and scalability compared to configurations without privacy measures.

KEYWORDS

blockchain, rubber supply chain management, traceability, sustainability, security, privacy-preserving

1 Introduction

Cambodia's Natural Rubber (NR) sector began with small-scale cultivation in the early 1900s and saw industrialization begin in 1921 (Hang, 2009). By the 1970s, NR covered about 60,000 ha (ha). The sector expanded significantly to 436,682 ha across 22 provinces by 2018, with a production capacity of 220,000 tons projected to grow from 2019 to 2023. The majority of NR processing and exports consist of 78% Technically Specified Rubber (TSR) and 21% Ribbed Smoked Sheets (RSS), with the remainder being Concentrated Latex (CL)[1].

1 https://gdr1.maff.gov.kh/?lang=en (Accessed 01-04-2024).

The Cambodia Rubber Supply Chain Management (RSCM) sector involves 284 participants, including estate agencies, manufacturers, and cooperatives (Diepart et al., 2023). Despite the absence of a rubber futures market, Cambodia relies on market-reference-based pricing (MRB), which is influenced by international markets such as TOCOM, SICOM, and Shanghai. Cambodia's NR export markets include Vietnam, Malaysia, China, Singapore, South Korea, and Europe.

In addition, agriculture, an important part of Cambodia's economy, contributed 22% to the GDP in 2022 and employed 2.6 million people[2]. This sector plays a critical role not only in economic output but also in supporting the livelihoods of a significant portion of the population, particularly in rural areas where farming is the primary source of income and employment. Despite its importance, Cambodia's agriculture, including the rubber industry in this context, faces several challenges, such as labor issues, unstable markets, and insufficient traceability. These challenges hinder the sector's growth potential and its ability to fully capitalize on international demand, especially in industries like rubber, where issues related to production quality and supply chain transparency are ongoing concerns. As a result, improving agricultural practices, enhancing market stability, and implementing better traceability systems could significantly contribute to the sustainable development of Cambodia's agricultural sector.

Motivated by the above issues, this paper aims to develop a decentralized data-sharing and storage system to enhance sustainability and traceability in Cambodia's rubber supply chain management. We leverage blockchain as a distributed ledger technology to provide a transparent and secure method for tracking rubber throughout the supply chain, thereby improving efficiency, fairness, and stakeholder confidence. In this context, we employ a consortium blockchain to establish a decentralized, community-driven governance structure that involves key stakeholders, including the Cambodian government, the private sector, NGOs, and local communities. This approach utilizes blockchain consensus mechanisms to ensure broad participation and transparency in decision-making. Moreover, blockchain technology can be deployed to create effective incentive mechanisms, using smart contracts and token-based rewards to encourage active participation and commitment from stakeholders. In addition, we use Hyperledger Fabric to provide a scalable, flexible framework for secure transactions, while the InterPlanetary File System (IPFS) ensures secure, decentralized data storage. Hyperledger Fabric's consortium-based architecture allows for controlled access and collaboration among trusted participants, making it an ideal solution for managing a network with multiple stakeholders while ensuring security, scalability, and privacy. Furthermore, we also incorporate differential privacy, which protects sensitive data while enabling meaningful data sharing. Differential privacy was chosen for this study due to its strong theoretical foundation and proven effectiveness in providing privacy guarantees while allowing for meaningful data analysis. Unlike other privacy-preserving methods such as k-anonymity or homomorphic encryption, which may face challenges like loss of data utility or high computational overhead, differential privacy provides a formalized framework that ensures that the inclusion or exclusion of any individual's data has minimal impact on the overall results, thus protecting privacy. The primary objectives of this system are to enhance traceability and visibility, promote sustainable practices, and secure sensitive business data. By achieving these objectives, the system aims to increase the competitiveness and viability of the natural rubber (NR) industry, thereby contributing to the overall economic growth and sustainability of Cambodia. We summarize the contributions of this paper as follows:

1. In this paper, we propose a conceptual framework designed to leverage blockchain technology for enhancing and modernizing agricultural distribution networks. We leverage blockchain to enhance pricing transparency and supply chain traceability with a decentralized approach. Moreover, by establishing a secure and tamper-proof record of transactions, our framework effectively reduces fraud and corruption, fostering trust among stakeholders and facilitating ethical sourcing practices. Additionally, it is important to note that the system described has not yet been implemented with real farmers, real fields, or real-time data from IoT devices. Instead, the focus is on illustrating how blockchain can be applied to create a more secure and transparent supply chain.
2. We utilize IPFS to facilitate a peer-to-peer distributed file system and provide effective off-chain storage. By leveraging IPFS, the paper ensures the protection of sensitive information and maintains data integrity, thereby enhancing the overall security framework of the system.
3. Additionally, we employ DP with the Laplace mechanism to safeguard sensitive data during the data-sharing process. This innovative approach guarantees the rigorous protection of individual privacy and data confidentiality while facilitating the sharing of crucial information among stakeholders. As a result, it significantly enhances the overall security and integrity of blockchain-based RSCM systems.

2 Background

2.1 Supply chain management

Supply chain management (SCM) focuses on the strategic synchronization of product information and data movements across organizations, aiming to deliver products or services to consumers efficiently. This includes coordinating operations such

Abbreviations: NR, Natural Rubber; ha, Hectare; TSR, Technically Specified Rubber; RSS, Ribbed Smoked Sheets; CL, Concentrated Latex; RSCM, Rubber Supply Chain Management; MRB, Malaysian Rubber Board; TOCOM, Tokyo Commodity Exchange; SICOM, Singapore Commodity Exchange; SHFE, Shanghai Futures Exchange; IoT, Internet of Things; IPFS, InterPlanetary File System; DP, Differential Privacy; HLF, Hyperledger Fabric; TPS, Transactions Per Second; MQTT, Message Queuing Telemetry Transport; API, Application Programming Interface; ETH, Ethereum; SC, Smart Contract; CIDs, Content Identifiers; CA, Certificate Authority; PoW, Proof-of-Work; PoS, Proof-of-Stake; VM, Virtual Machine; JVM, Java Virtual Machine; DLT, Distributed Ledger Technology; CFT, Crash Fault Tolerant.

2 https://www.fao.org/hand-in-hand/hih-IF-2023/cambodia/en (Accessed 01-04-2024).

as raw material purchasing, manufacturing planning, logistics, inventory management, and forecasting consumer demand. Effective SCM is essential for cost savings, market responsiveness, and consumer satisfaction (Kumar and Sahoo, 2025). Organizations must balance trade-offs, like cost reduction and maintaining sufficient supply to meet demand. The complexity of global supply chains necessitates advanced SCM practices utilizing technologies like blockchain, data analytics, AI, and IoT for real-time visibility and agility (Ivanov et al., 2019). Traditional SCM also emphasizes sustainability, reducing environmental impact, and ensuring transparency in purchasing processes, reflecting rising consumer awareness and regulatory demands.

2.1.1 Cambodia rubber value chain

In Cambodia's rubber value chain, latex is processed into rubber using RSS or TSR methods. Smallholder farmers sell latex or coagulum based on factors like purchase price, processing unit proximity, and transportation availability. Numerous RSS processing facilities exist near smallholder farms, accessible within a day by motorcycle. Large landholders may sell directly to RSS facilities, exporting processed rubber to destinations such as Korea, Vietnam, China, India, and Malaysia (Diepart et al., 2023). TSR processing units also operate, sourcing latex and coagulum, with primary exports to Eastern Europe, Vietnam, China, Korea, Malaysia, and India. Smallholder farmers often coagulate latex on the farm and sell to local collectors, who transport it directly to Vietnam and Thailand, bypassing RSS or TSR processing. Supply chain management (SCM) focuses on the comprehensive synchronization of product information and data movements throughout an organization of businesses, including the primary objective of delivering products or services to the consumer or end user. Subsequently, it represents strategic coordination of operations such as raw material purchasing, manufacturing planning, logistics, inventory management, and forecasting consumer demand. Effective SCM is fundamental to achieving cost savings, increasing market responsiveness, and improving consumer endorsement (Kumar and Sahoo, 2025).

2.2 Blockchain technology

The concept of Blockchain (BC) has significantly changed our perception of reliability and cooperation in distributed applications (DApps). It is based on a shared, indestructible ledger made of cryptographically attributed components Firdaus et al. (2023a). Satoshi Nakamoto's 2008 Bitcoin whitepaper mentioned this revolutionary invention (Nakamoto, 2024). In Figure 2, each record has a header with versioning, timestamps, and a Merkle root for data confidentiality, and an object containing transaction details. Blockchains achieve immutability through hash pointers, with each block referencing its predecessor for secure traceability. These ledgers operate on decentralized networks, relying on consensus mechanisms like Proof-of-Work (PoW) to validate updates (Cachin and Vukolić, 2017) and cryptography to protect data during transfer (Pappachan et al., 2024). This design fosters decentralization, immutability, traceability, collaborative maintenance, and customizable transparency (Pilkington, 2016). However, Bitcoin's pioneering protocol exposes constraints in scalability and its scripting language's adaptability for complex applications (Croman et al., 2016).

2.2.1 Blockchain platforms

1. Ethereum: Ethereum is a decentralized system that allows smart contracts to function. Ethereum creates machine code from smart contracts, which are then run by the Ethereum virtual machine (EVM) (Wood et al., 2014). Ethereum smart contracts use an account-based data structure where each user is identified by their digital wallet (Antonopoulos and Wood, 2018). Ethereum employs the computationally costly PoW consensus process, just like Bitcoin (Nakamoto, 2024). However, Proof-of-Stake (PoS) will soon replace PoW on Ethereum, expanding Ethereum through *E*th2 enhancements (Buterin et al., 2014). Gas functions as an internal charge for executing a transaction to offset ETH's unstable value (Wood et al., 2014). Ether, the native cryptocurrency token of Ethereum, provides an engine for transaction execution and ecosystem-wide interaction with DApps (Antonopoulos and Wood, 2018). Efforts to address scalability challenges are motivated by modest transaction latency and current scalability restrictions, assessed at 15 transactions per second (TPS) (Zheng P. et al., 2018). The large Ethereum development community continuously optimizes the protocol and contributes to regular upgrades (Buterin et al., 2014).
2. Hyperledger Fabric: Hyperledger Fabric, a distributed ledger technology (DLT) within the Hyperledger project overseen by the Linux Foundation[3], offers a distinct approach to smart contract execution. Unlike Ethereum's reliance on a Virtual Machine (EVM), Fabric leverages Docker containers for smart contract code deployment. This strategy provides enhanced isolation and resource efficiency compared to VMs (Shalaby et al., 2020). While initially receiving substantial investment from IBM, Fabric's open-source nature promotes collaboration and prevents single-entity dominance. Fabric supports traditional high-level languages like Java and Go, contrasting with Ethereum's domain-specific languages (Vukolić, 2016). Fabric maintains Turing completeness, ensuring the network's ability to perform general-purpose computations. Fabric employs a key-value data model for state representation. Designed for enterprise deployment, Fabric implements a permissioned blockchain model, requiring network participation authorization from Certificate Authorities (CAs) (Androulaki et al., 2018). Multiple CA types serve distinct roles within the network. Fabric's permissioned structure streamlines consensus mechanisms, optimizing for efficiency within controlled environments (Sousa et al., 2018).
3. R3 Corda: Corda focuses on use cases addressing digital currencies and assets, providing a framework for managing and documenting digital asset ownership[4]. High-level

3 https://hyperledger-fabric.readthedocs.io/en/release-2.5/ (Accessed 06-04-2024).

4 https://www.r3.com/white-papers/corda-technical-whitepaper/ (Accessed 08-04-2024).

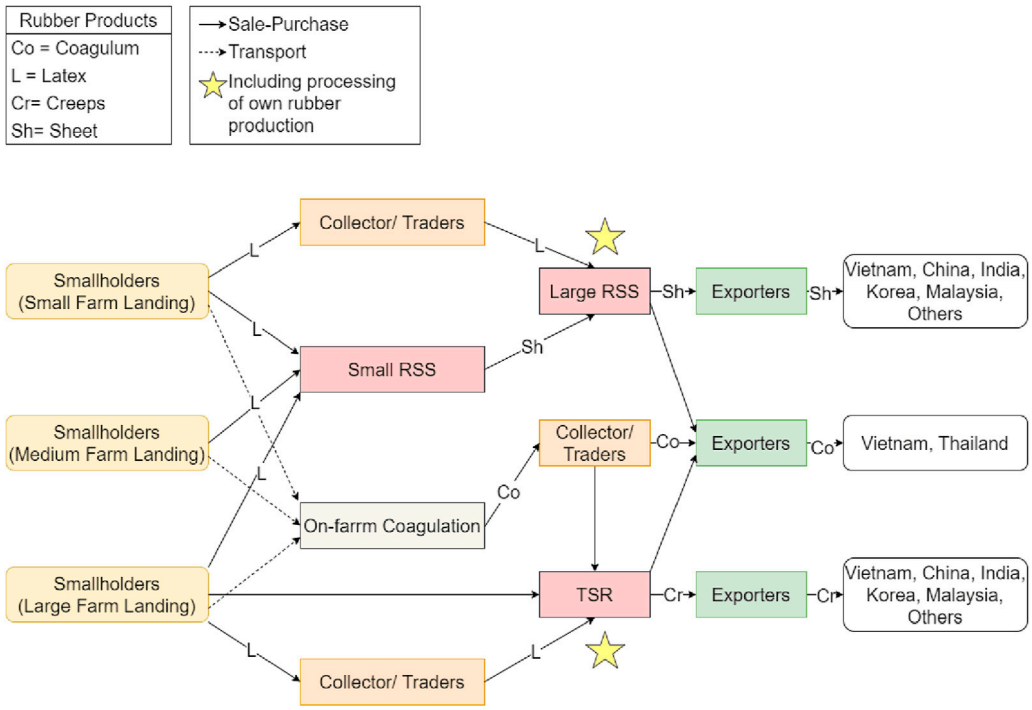

FIGURE 1
The general structure of Cambodia's Rubber Supply Chain. [Source: Author, adapted from (Diepart et al., 2023)].

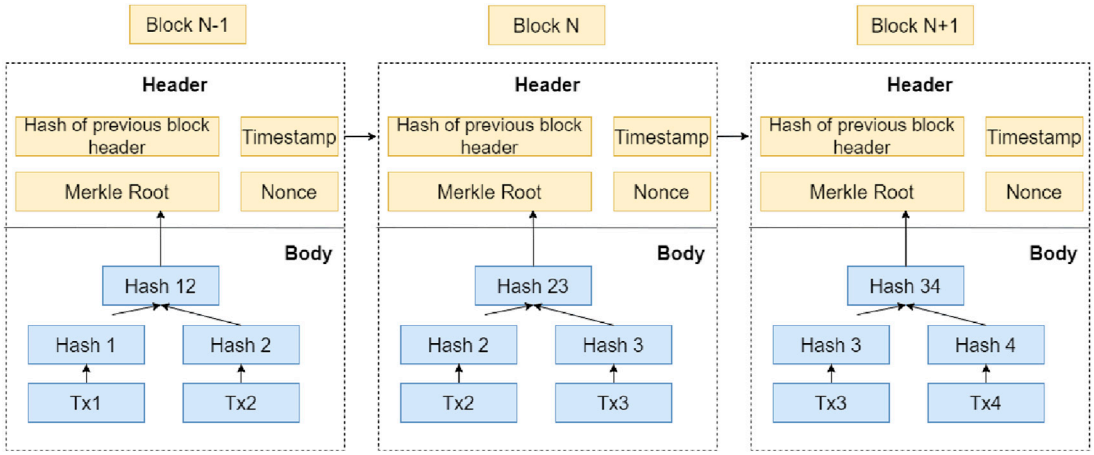

FIGURE 2
Structure of the blocks (Zheng et al., 2017)

programming languages like Java and Kotlin are used to write Corda's smart contracts, running in the Java Virtual Machine (JVM). Corda confirms transaction validity only among immediate parties involved, not throughout the network. With a focus on asset statuses and their changes, Corda uses a transaction-based data architecture, often resulting in its deployment in private, permissioned blockchains (Belchior et al., 2021). Corda's permissioned structure and use of the RAFT consensus method speed up consensus achievement (Ongaro and Ousterhout, 2014). RAFT operates as a "leader and follower" ordering service, with decisions replicated by a predetermined leader node. It uses a crash-fault-tolerant (CFT) architecture, guaranteeing consensus finality and protocol protection even if some network components fail (Cachin and Vukolić, 2017).

2.2.2 Smart contract in hyperledger fabric

Smart contract (SC) are the core foundation behind Hyperledger Fabric's efficiency and performance. In this permissioned blockchain architecture, SC operates as self-executing contracts,

FIGURE 3
Smart contracts in HLF.

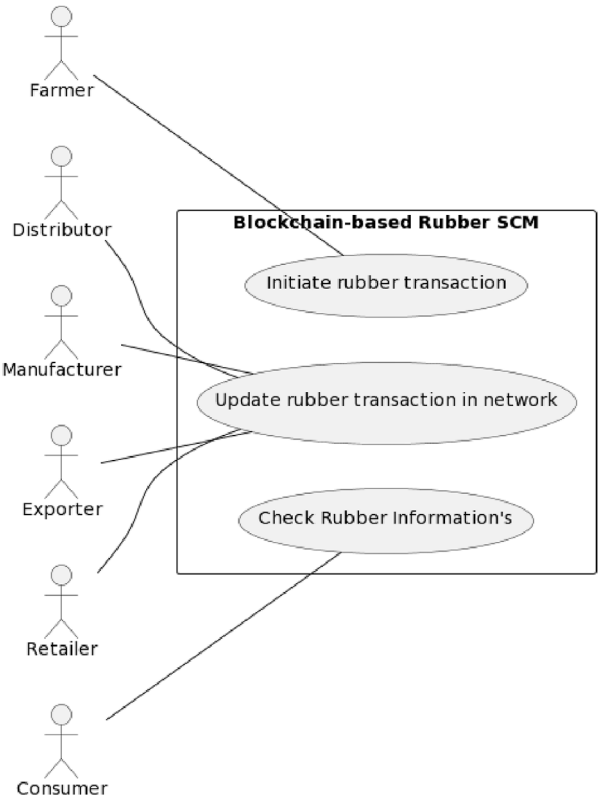

FIGURE 4
Rubber SCM based on Blockchain.

including agreed-upon business logic directly throughout the source code (Androulaki et al., 2018). Data preservation, in addition to an immutable record of state change transactions, provides remarkable transparency and automation to operations previously limited by complications, inefficiencies, and possible conflicts (Christidis and Devetsikiotis, 2016). In Hyperledger Fabric, SC is arranged into components known as Chain Code (CC) for administrative ease (Androulaki et al., 2018). These CCs, primarily created in Go but also support Node. js and Java, are run by network peers. This peer execution ensures that all authorized parties agree on transaction results before modifying the ledger state (Vukolić, 2016) as shown in Figure 3.

2.2.3 IPFS with hyperledger fabric

The InterPlanetary File System (IPFS) offers an attractive solution for blockchain technology's data storage limitations. IPFS, a peer-to-peer (P2P) distributed file system, uses content-addressing to generate unique content identifiers (CIDs) through file hashing. By storing these CIDs on the blockchain instead of raw data, IPFS significantly reduces storage pressure and costs (Zheng Q. et al., 2018; Firdaus et al., 2023b). This relationship between IPFS and blockchain allows effective off-chain storage of significant or sensitive data. Sensitive information benefits from IPFS's distributed nature, enhancing privacy, while the blockchain provides a tamper-proof record for references and metadata. Furthermore, IPFS incentivizes persistent data storage via Filecoin, its native asset, ensuring ongoing accessibility (Xu et al., 2018). Additionally, the content-addressable aspect of IPFS presents various benefits. Immutability is achieved by identifying content via a unique hash; any changes to the file's contents generate a new CID. This improves version management, maintains verifiable data connections, and develops trust in data consistency. Therefore, integrating IPFS with blockchain systems allows decentralized data management, reducing dependency on centralized storage providers while offering a more secure and flexible structure.

2.3 Related works

Blockchain technology has shown significant potential in transforming agriculture and supply chain management by

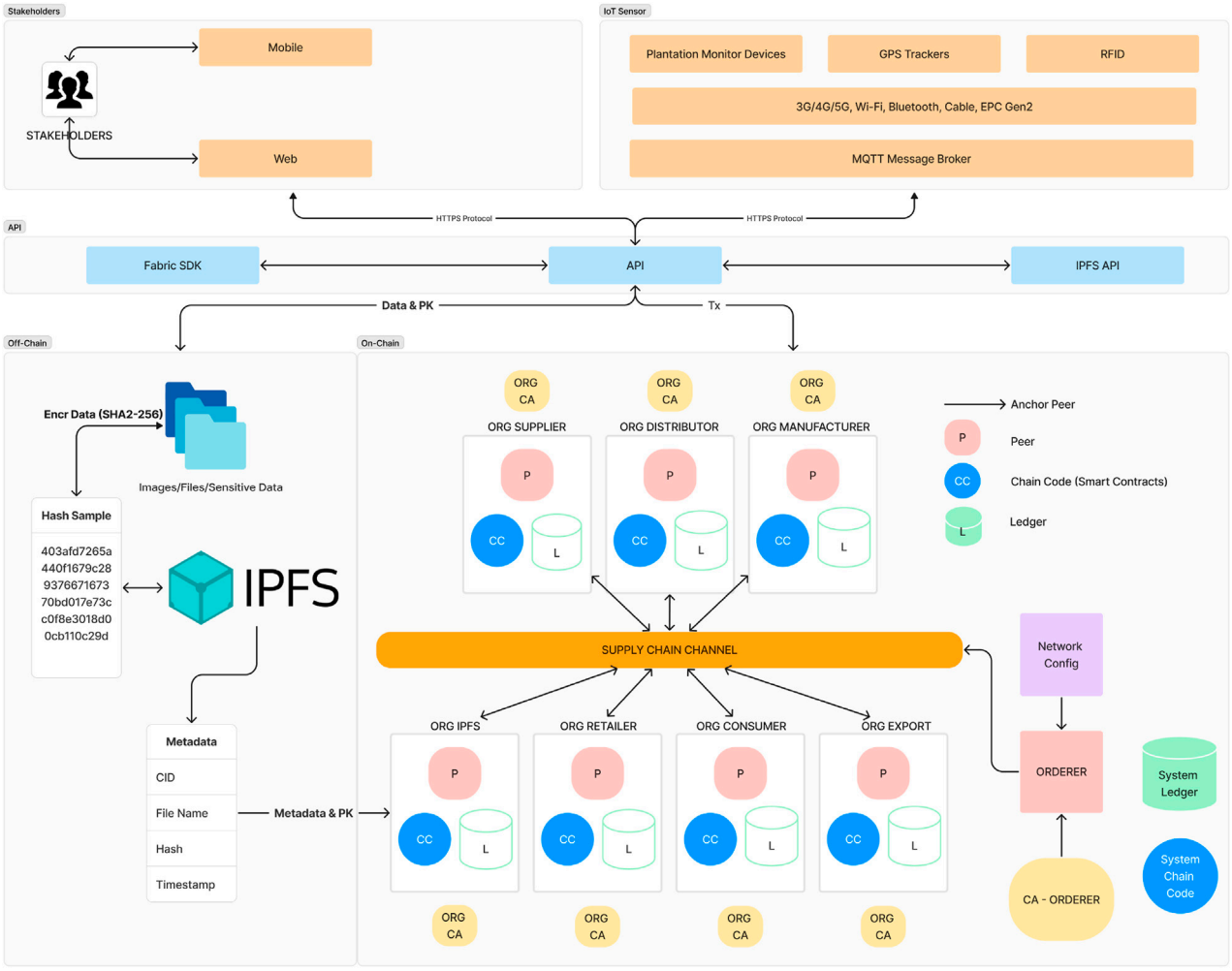

FIGURE 5
Details proposed system network configurations based on HLF-BC and IFPS for Cambodia RSCM.

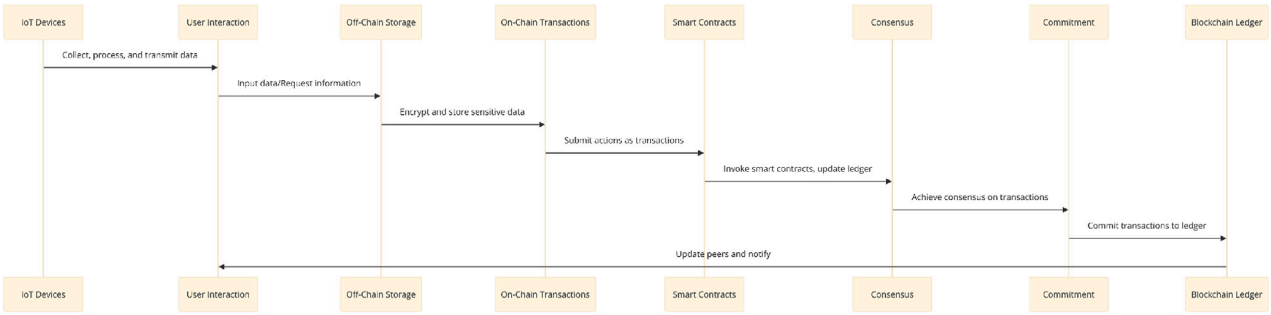

FIGURE 6
Transaction flow in rubber BC-based SCM.

enhancing transparency, traceability, and operational efficiency. Several studies have explored its application across diverse agricultural domains and specific commodity supply chains like rubber. Blockchain has been widely applied in general agricultural supply chains to address issues such as food safety, fraud, and inefficiency. Tian (2016) introduced a blockchain-based food traceability system, enabling real-time monitoring of agricultural products, thus improving food safety. Kamilaris and Fonts, 2019

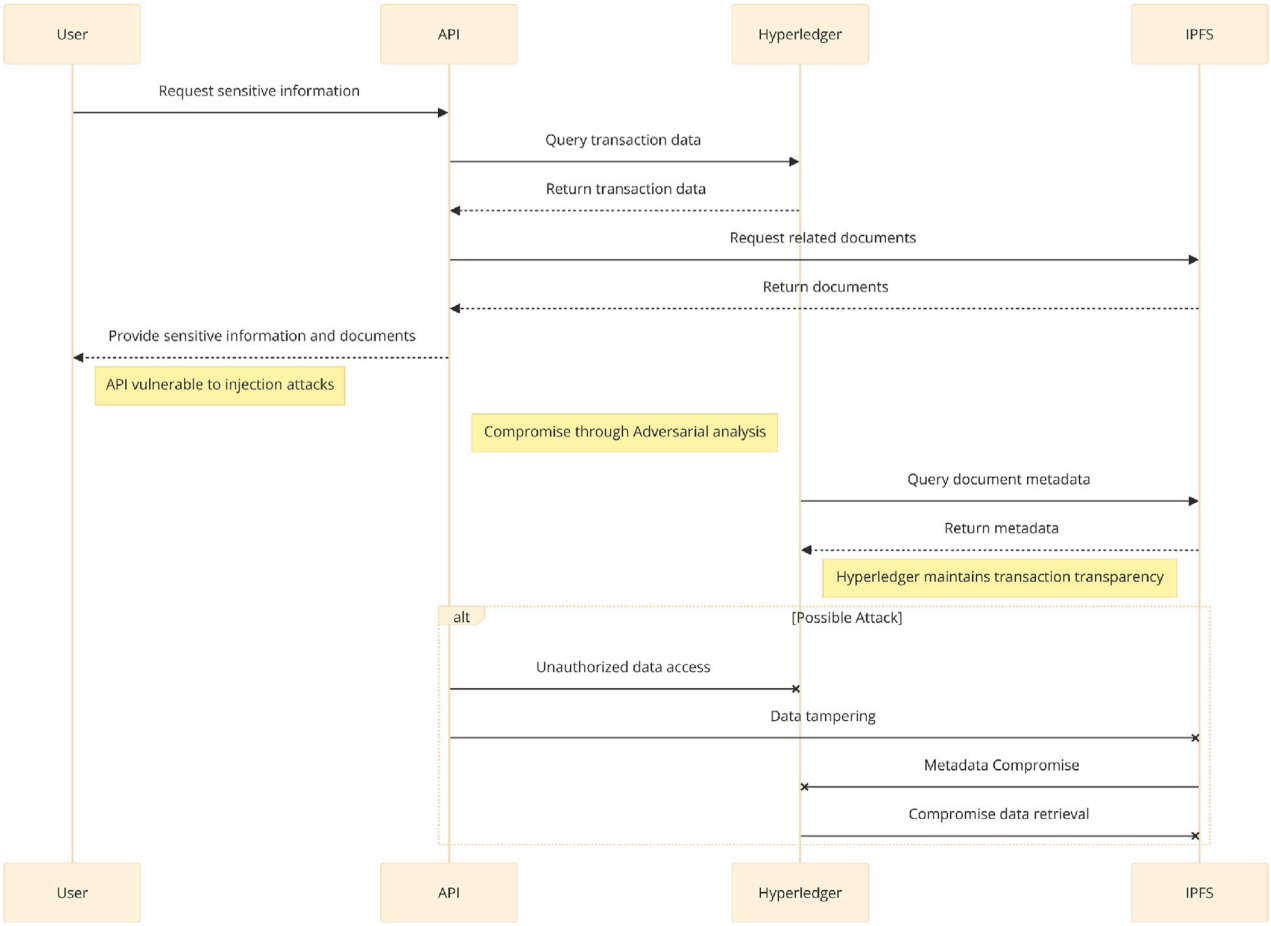

FIGURE 7
The scenario of compromise sensitive information in rubber SCM Blockchain.

provided a comprehensive review of blockchain applications in agriculture, highlighting its role in improving supply chain transparency and combining it with IoT for efficient data collection and sharing. Similarly, Tripoli and Schmidhuber (2018) discussed blockchain's role in agricultural trade by enabling farmers to access markets directly, reduce intermediaries. In sustainable agriculture, Dey and Shekhawat (2021) examined blockchain's ability to track and verify sustainable farming practices, providing greater accountability to consumers and stakeholders. Using blockchain to manage smart farming systems with IoT integration has also been explored by Caro et al. (2018), who developed a system to automate crop monitoring and optimize supply chain workflows.

The rubber supply chain faces unique challenges, including ensuring product authenticity, addressing deforestation concerns, and promoting sustainable sourcing. Blockchain solutions have been proposed to improve transparency and mitigate these issues. For example, Yadav et al. (2024) developed a blockchain-enabled traceability framework for the natural rubber supply chain, allowing real-time monitoring of sourcing and production practices while ensuring adherence to sustainability standards. In a similar context, Cui and Gaur (2022) explored blockchain's role in supporting smallholder farmers by integrating rubber price tracking and smart contracts to facilitate fair payments. Their study showed how blockchain reduces exploitation and promotes equitable profit sharing among stakeholders. Furthermore, Xiong et al. (2020) demonstrated how smart contracts can be applied in rubber trading to automate payment processes and ensure contract compliance, thus enhancing operational efficiency and reducing disputes.

3 Proposed framework

3.1 System overview

In this section, we present our proposed system to address the challenges faced by the rubber supply chain in Cambodia, including limited traceability, poor visibility, potential security risks, and concerns about sustainability. To overcome these issues, we leverage HLF's flexibility and suitability for complex supply chains. Moreover, HLF is chosen for the rubber supply chain management system due to its unique advantages in scalability, performance, and privacy, which align with the requirements of the domain. Its permissioned architecture ensures that only authorized participants, such as producers, manufacturers, and distributors, can

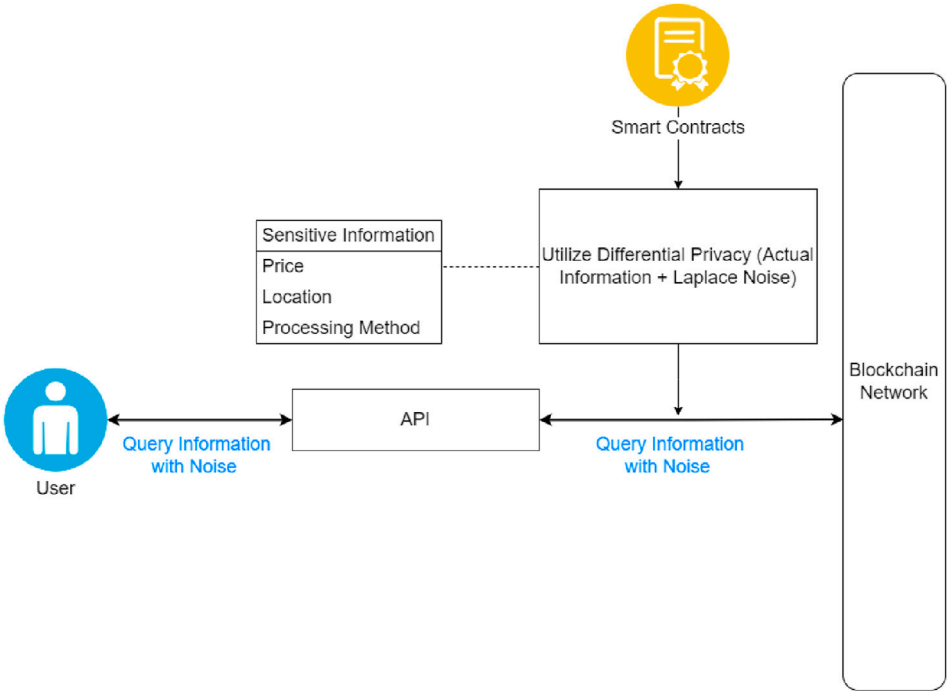

FIGURE 8
The scenario of compromise sensitive information in rubber SCM Blockchain.

access and validate transactions, enhancing security and trust within the network. Fabric's modular design supports channel partitioning and private data collections, allowing sensitive supply chain data to be shared securely among relevant stakeholders while maintaining privacy. Furthermore, its pluggable consensus mechanism is well-suited in the rubber industry. To securely store sensitive information such as rubber prices, origin details, and processing methods, we incorporate IPFS as an off-chain storage solution. In addition, IoT devices play a vital role in real-time monitoring, utilizing communication protocols like Bluetooth, WiFi, and the Internet to collect and transmit data. Protocols such as MQTT, HTTPS, and APIs integrate IoT devices with the blockchain and user interfaces, ensuring a seamless flow of data for effective supply chain management. To ensure secure data sharing, we employ differential privacy based on the Laplace mechanism, integrated into the smart contract, to protect sensitive information while maintaining data utility.

3.1.1 System entities

Our proposed system includes seven organizations, including Farmers, Distributor, Manufacturers, Exporters, Retailers, Consumers, and IPFS, as described in the following paragraphs and Figures 4, 5.

- Farmer Organization: Farmer organizations provide the basis for cultivating rubber trees, harvesting latex, and processing it into useful forms such as bales or sheets. They start the rubber's journey by connecting it to the supply chain network.
- Distributor Organization: Distributors act as intermediaries between manufacturers and Farmers. They assure a continual supply of materials throughout the production process by purchasing raw rubber from Farmers, overseeing its storage, and distributing it to manufacturers according to their demands.
- Manufacturer Organization: To make a wide variety of rubber goods, such as tires, hoses, and seals, they buy raw rubber from distributor organizations. They provide value through manufacturing processes by converting raw resources into completed commodities.
- Exporter Organization: Exporters enable commerce between countries. By acquiring completed rubber goods from producers, managing export paperwork and shipping, and guaranteeing that the goods get to global consumers, they help the rubber sector grow internationally.
- Retailer Organization: Retailers are the last port of call. Completed rubber goods are acquired from exporters or distributors and sold to customers via a variety of channels, including retail locations and internet retailers. They bring the finished product to the people who use rubber items on a daily basis, serving as the last point of sale.
- Consumer Organization: Retailers are the last port of call. Completed rubber goods are acquired from exporters or distributors and sold to customers via a variety of channels, including retail locations and internet retailers. They bring the finished product to the people who use rubber items on a daily basis, serving as the last point of sale.
- IPFS Organization: The IPFS, a decentralized and impenetrable network platform, is run by the IPFS group. This system promotes openness and confidence across the supply chain by enabling the safe storage and

```
hlf1@hlf1-vm:~/Rubber_Supply_Chain$ minifab up
Using default spec file
# Running operation: ****************************************
  channel join
.........................
# Running operation: ****************************************
  anchor update
...........
# Running operation: ****************************************
  profile generation
.............................
# Running operation: ****************************************
  cc install
.........................
# Running operation: ****************************************
  cc approve
......
# Running operation: ****************************************
  cc commit
............
# Running operation: ****************************************
  cc initialize
......
# Running operation: ****************************************
  discover
......................
# Discover endorsers results ********************************
  Chaincode endorsers file: ./vars/discover/mychannel/simple_endorsers.json
# Discover orderers results *********************************
  Channel orderers file: ./vars/discover/mychannel/ordererendpoints.json

# STATS *****************************************************
minifab: ok=431 failed=0

real    6m34.981s
user    2m40.578s
sys     0m46.542s
```

FIGURE 9
Network initialization logs.

exchange of critical rubber-related data, including contracts, certifications, and sensor data.

3.1.2 System architecture

- Stakeholder Engagement: Initially, stakeholders, including Farmers, manufacturers, and retailers, interact with the system via mobile and web interfaces. These interfaces are designed for intuitive use, allowing stakeholders to input data, retrieve information, monitor processes, and make informed decisions efficiently. Moreover, this user-friendly approach facilitates widespread accessibility and constant communication, crucial for real-time supply chain management.
- IoT Sensor Integration: Subsequently, the system incorporates IoT devices such as plantation monitor devices, GPS trackers, and RFID sensors. These devices are instrumental in collecting real-time data and tracking the movement of goods across the supply chain. They connect using various protocols, including 3G/4G/5G, Wi-Fi, and Bluetooth, ensuring comprehensive coverage and connectivity. Data collected from these devices are typically transmitted via an MQTT message broker, which is a standard protocol for IoT communications, enhancing the reliability and timeliness of data transmission.
- Blockchain Interaction: Moreover, the core of the system utilizes Hyperledger Fabric, a permissioned blockchain framework known for its robust security features and performance efficiency. Interaction between the users and IoT devices with the blockchain is facilitated through API interfaces using the Fabric SDK. All transactions are conducted over HTTPS to ensure secure and reliable data transmission. This setup ensures that all interactions within the supply chain are immutable and verifiable, enhancing trust among all participants.
- Data Management: The integration of HLF in the rubber supply chain offers a secure, transparent, and efficient solution to address key data management issues. Through role-based access control (RBAC), it ensures data governance by granting permissioned access to authorized participants only, safeguarding sensitive information. The blockchain's immutability provides full traceability of rubber products,

```
hlf1@hlf1-vm:~/Rubber_Supply_Chain$ minifab create -c rubbersupplychain
Using default spec file
Minifab Execution Context:
    FABRIC_RELEASE=2.3.0
    CHANNEL_NAME=rubbersupplychain
    PEER_DATABASE_TYPE=golevel
    CHAINCODE_LANGUAGE=go
    CHAINCODE_NAME=simple
    CHAINCODE_VERSION=1.0
    CHAINCODE_INIT_REQUIRED=true
    CHAINCODE_PARAMETERS="init","a","200","b","300"
    CHAINCODE_PRIVATE=false
    CHAINCODE_POLICY=
    TRANSIENT_DATA=
    BLOCK_NUMBER=newest
    EXPOSE_ENDPOINTS=false
    CURRENT_ORG=org0.example.com
    HOST_ADDRESSES=192.168.119.129
    TARGET_ENV=DOCKER
    WORKING_DIRECTORY: /home/hlf1/Rubber_Supply_Chain
.....
# Preparing for the following operations: ********************
  verify options, channel create
..................
# Running operation: ****************************************
  verify options
..
# Running operation: ****************************************
  channel create
......

# STATS ******************************************************
minifab: ok=35  failed=0

real    0m12.616s
user    0m10.212s
sys     0m1.878s
```

FIGURE 10
Channel configuration logs.

allowing stakeholders to track the product's journey from farm to end-user and verify its compliance with quality standards. In case of disputed transactions, smart contracts enable automated, transparent resolution based on predefined rules. On the other hand, backup and recovery procedures are integrated with on-chain and off-chain solutions, ensuring data integrity and enabling quick restoration in the event of system failures.

- Off-Chain Data Storage: For managing sensitive data such as images and confidential documents, the system employs SHA-256 encryption and stores this data off-chain in the IPFS. This approach prevents the blockchain from being overwhelmed by large volumes of data while ensuring that the data remains accessible and secure. Metadata for these files, along with content identifiers (CIDs), are anchored to the blockchain, maintaining the integrity and traceability of off-chain stored data.
- On-Chain Process Flow: Within the blockchain, various organizational entities interact through dedicated channels termed 'SUPPLY CHAIN CHANNEL'. These interactions involve peers (P), chaincode (CC), and ledgers (L) for different entities such as Farmers, distributors, and consumers. This delineation facilitates efficient and secure transaction processing and information flow within the network.
- Implementation Constraints: We propose a hybrid solution to overcome the dependency on stable internet connectivity in rural Cambodia, allowing critical transactions to be conducted offline and synchronized once connectivity is restored. Low-bandwidth optimization techniques such as data compression and transaction batching will also be employed to minimize network demands, with mesh networking explored for intermittent connectivity. To tackle hardware limitations, we recommend using lightweight devices like smartphones or IoT sensors for small-scale farmers, while offloading intensive computations to more powerful network nodes or cloud services. Additionally, to ensure long-term sustainability, a maintenance and upgrade framework will be established, including regular system audits, software updates via over-

```
fabric:
  cas:
    - "ca.supplier"
    - "ca.distributor"
    - "ca.manufacturer"
    - "ca.exporter"
    - "ca.retailer"
    - "ca.consumer"
    - "ca.ipfs"
  peers:
    - "peer.supplier"
    - "peer.distributor"
    - "peer.manufacturer"
    - "peer.exporter"
    - "peer.retailer"
    - "peer.consumer"
    - "peer.ipfs"
  orderers:
    - "orderer.rubbersupplychain"
  settings:
    ca:
      FABRIC_LOGGING_SPEC: DEBUG
    peer:
      FABRIC_LOGGING_SPEC: DEBUG
    orderer:
      FABRIC_LOGGING_SPEC: DEBUG

  netname: "rubbersupplychain"
```

FIGURE 11
Organization COnfiguration logs.

the-air mechanisms, and local technical support teams to assist farmers with troubleshooting and hardware maintenance. This approach ensures accessibility, functionality, and continued evolution of the system in alignment with farmers' needs.

- Network Configuration: Lastly, the network configuration of the system is designed to be modular. Components such as the ORDERER play a crucial role in the consensus and ordering of transactions into blocks. The System Channel Code manages network-wide settings, and the System Ledger records the state of the network, with an anchor peer serving as a synchronization point for organizational data.

3.2 Transaction flow

The proposed Hyperledger Fabric-based rubber supply chain management system in Cambodia incorporates a sophisticated architecture designed to enhance the integrity, efficiency, and transparency of the supply chain. It leverages IoT technology to collect essential data and employs blockchain technology for secure data processing and storage. By integrating these technologies, the system aims to address the traditional challenges of supply chain operations, such as limited traceability and potential security risks. The initial stage of the system architecture involves extensive data collection via IoT sensors strategically placed throughout the supply chain. These sensors gather critical real-time data, including temperature readings from storage facilities, humidity levels crucial for rubber preservation, and GPS tracking information to monitor logistical movements. This data is vital for proactive supply chain management, enabling immediate responses to potential issues, such as temperature fluctuations that could compromise product quality. To ensure secure and efficient data transmission, the system employs MQTT (Message Queuing Telemetry Transport), a lightweight machine-to-machine (M2M) connectivity protocol optimized for low-bandwidth environments.

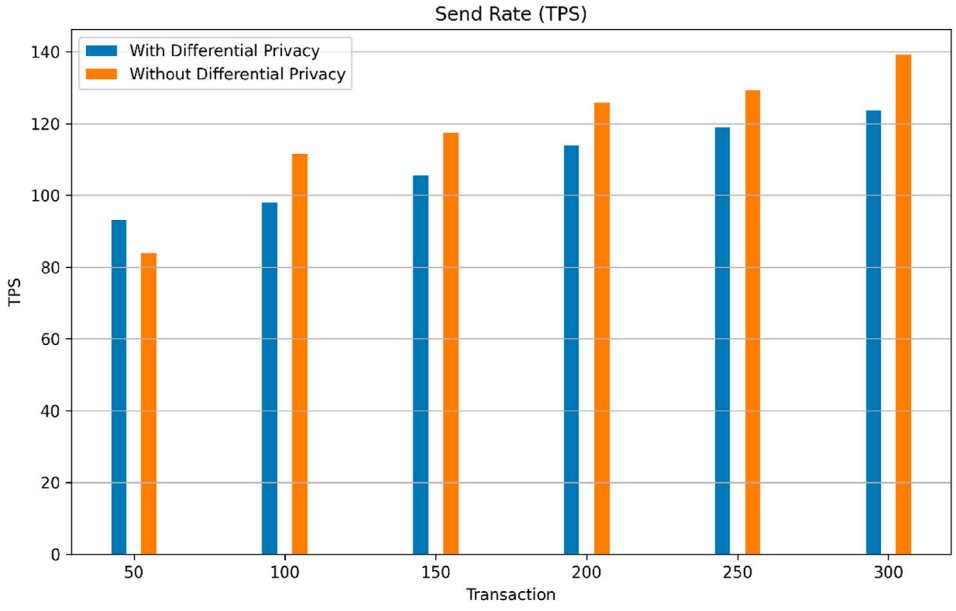

FIGURE 12
The Transaction send rate.

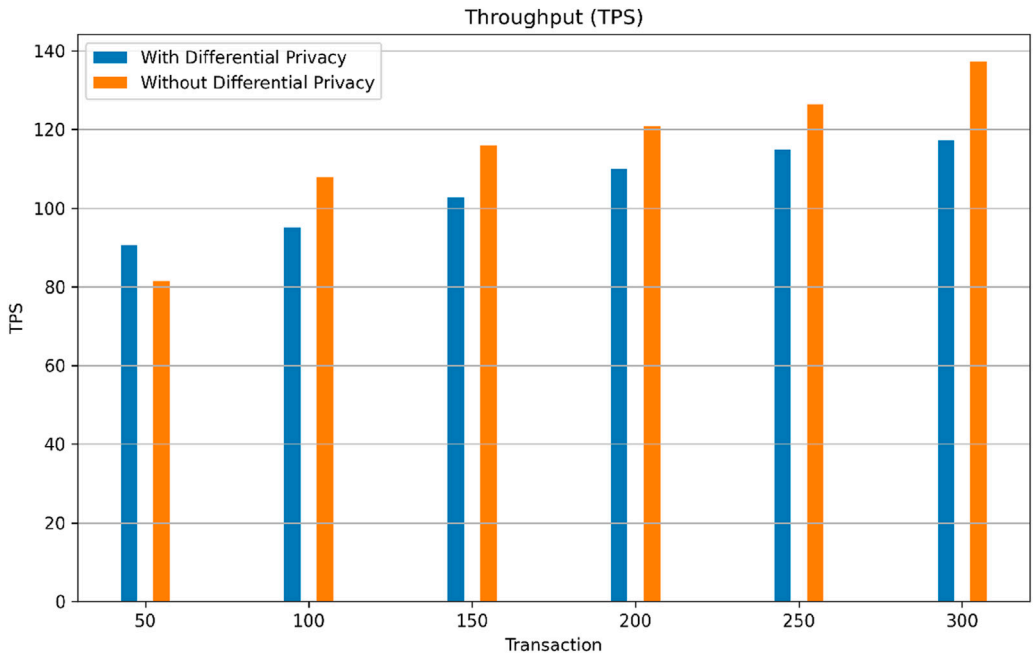

FIGURE 13
The Transaction throughput.

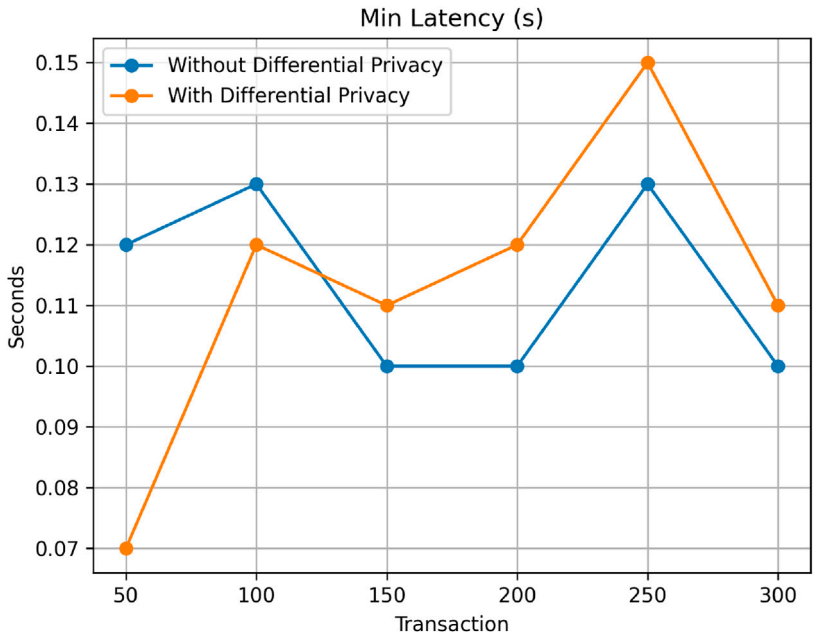

FIGURE 14
The minimum Transaction latency.

MQTT facilitates reliable communication between IoT devices and the central system, minimizing the risk of unauthorized access and data breaches. The collected data is securely processed and made available for further analysis and decision-making. Following data collection and processing, the system supports active stakeholder interaction through user-friendly mobile and web interfaces. These interfaces serve as the primary tools for data input and retrieval, enabling stakeholders to perform tasks such as entering shipment details, checking inventory levels, and monitoring supply chain operations in real time. Designed with accessibility in mind, these interfaces ensure that users of varying technical proficiency can effectively interact with the system.

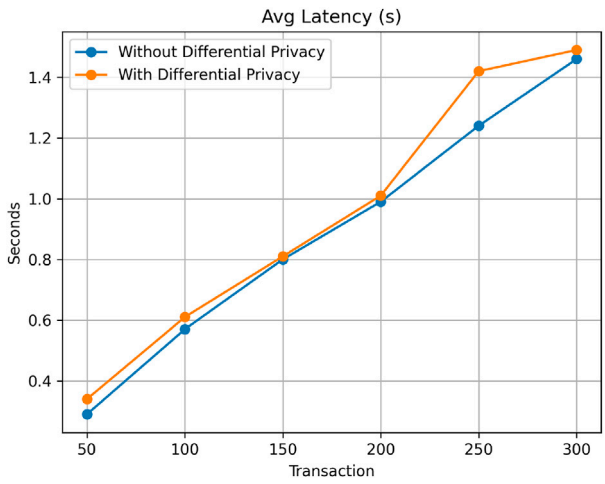

FIGURE 15
The average Transaction latency.

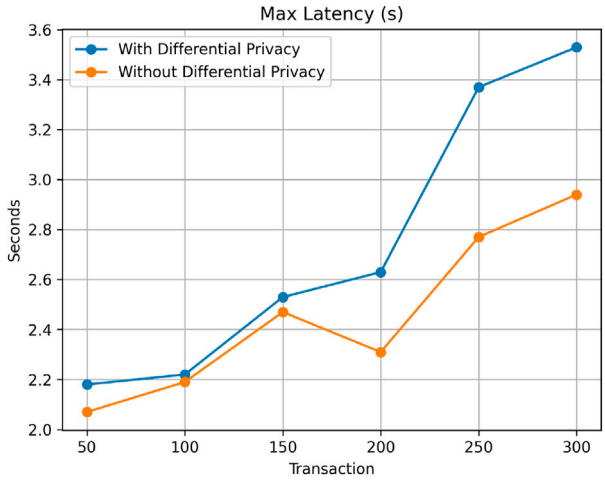

FIGURE 16
The maximum Transaction latency.

TABLE 1 Simulation parameters settings.

Parameter	Value
Operating System	Ubuntu 20.04.1 LTS 64-bit
CPU	Intel® Core™ i7-7700 8 CPU @ 3.60 GHz
RAM	8 GB
Minifab for Hyperledger Fabric	2.5 (Latest)
IPFS	12
Hyperledger Caliper	2.0
Docker-compose	1.29.2
Docker	25.0.3
Go	1.20.2
Google Differential Privacy Library (Go)	1.1.2

To enhance security, sensitive data such as personal information and logistical details are encrypted using SHA-256, a cryptographic hash function that provides robust data security. This encrypted data is stored off-chain on the InterPlanetary File System (IPFS), a peer-to-peer distributed file storage protocol. IPFS is ideal for handling large data volumes without compromising blockchain performance, as only metadata and hashes are stored on-chain. This approach reduces the blockchain's load while ensuring data immutability and availability. Central to the transaction flow is the creation and management of on-chain transactions. Stakeholder actions, such as placing orders or updating shipment statuses, are captured as transactions within the Hyperledger Fabric network. These transactions are managed via APIs that interface with the Hyperledger Fabric SDK, providing a secure and standardized method for transaction handling. Upon initiation, transactions trigger chaincode (smart contracts) programmed with the business logic of supply chain operations. The chaincode validates the transactions against predefined rules and ensures compliance with operational standards and regulatory requirements.

For transaction validation and ledger updates, the system employs the RAFT consensus mechanism. In this mechanism, a leader node is elected to propose and manage transaction ordering. This leadership and consensus approach ensures the integrity and trustworthiness of the system, as all nodes must agree on transaction data before committing it to the ledger. Once consensus is achieved, the transactions are committed to the blockchain, maintaining a chronological and immutable record of supply chain activities. To keep all participants synchronized, the system includes notification and synchronization mechanisms. After a transaction is committed, relevant stakeholders are notified, and their respective ledgers are automatically updated to reflect the latest state. This ensures data integrity across the network and enhances the visibility and reliability of supply chain operations. Through this detailed architecture and workflow, the proposed system harnesses the combined strengths of IoT and Hyperledger Fabric to revolutionize rubber supply chain management in Cambodia. By addressing traditional challenges of transparency, efficiency, and security, the system provides a robust solution that modernizes and streamlines supply chain operations.

3.3 Consensus protocol

The RAFT consensus mechanism is implemented in our proposed system to manage the ordering of transactions within our network. Distributed nodes use a leader-based consensus mechanism known as RAFT, or the Reliable, Replicated, Redundant, and Fault-Tolerant protocol, to ensure data consistency throughout the network. All client requests are processed by a leader, who monitors the log replication procedure and provides the data committed to the replicated log before execution. A new leader is chosen when the current one collapses, reducing the time needed to achieve consistency and

TABLE 2 Comparing the proposed framework with other related works.

Key features	Ref. #1	Ref. #2	Ref. #3	Ref. #4	Ref. #5	Ref. #6	Our work
Accessibility	×	✓	✓	×	✓	✓	✓
Data Security	×	×	✓	×	✓	×	✓
Fairer prices	✓	×	×	×	✓	✓	✓
Platform	N/A	N/A	ETH	N/A	ETH	N/A	HLF
Cost Reduction	×	×	✓	×	×	×	✓
Sustainability	✓	✓	✓	✓	×	✓	✓
Traceability	×	×	✓	✓	✓	×	✓
Transparency	×	×	✓	×	✓	×	✓
Use case	Rice	Bioenergy	Coffee	Rubber	Rubber	Rubber	Rubber

efficiently handle redundancy. RAFT is the most commonly utilized consensus mechanism in HLF implementations. It is very flexible and supports versions 2.0.0 and higher, which makes it suitable for our system, which is now running v2.5.0.

3.4 Secured data sharing protocol based on differential privacy using laplace algorithms

3.4.1 Vulnerable scenario in the HLF-based network

Figure 7 illustrates a possible vulnerable scenario in our proposed system for data exchange between the user, API, HLF, and the IPFS. This scenario describes a sequence of actions an attacker could employ to compromise system security. The initial vulnerability was identified at the API level, where the user enters sensitive information. The API is vulnerable to injection attacks, a type of manipulation in which an attacker injects malicious code into the system, resulting in unauthorized access or data tampering. This breach might compromise sensitive user information and allow the attacker to gain further capabilities within the network. Additionally, the adversarial assessment indicates that Hyperledger, regardless of its transaction transparency, is vulnerable to possible security breaches. The scenario exposes illegal data access, data manipulation, and metadata breaches. Each of these attack routes presents a critical risk to the system's security. Unauthorized access could expose confidential transaction data; data tampering could alter transaction records, compromising the ledger's reliability; and metadata compromise could reveal more details that could be used in future attacks or to cause current compromises (Islam et al., 2024). Furthermore, utilizing IPFS to query document information might offer an attack surface. However, rather than directly shown in the illustration, the metadata transmission could be intercepted or manipulated, resulting in a compromised retrieval of the information system.

3.4.2 Differential privacy (DP): a solution for the data sharing in HLF-based network

In our research, we propose leveraging Differential Privacy (DP) to enhance data security, privacy, and efficiency for rubber farmers. DP ensures that individual farmers' data remains confidential while still allowing for meaningful data analysis. It achieves this by injecting noise into the data before sharing or processing it, preventing unauthorized parties from identifying specific individuals. DP enables stakeholders to analyze rubber production trends without exposing individual farmers' private data (e.g., yield, pricing, and financial records). Moreover, DP helps mitigate various attacks, particularly membership inference attacks, which are crucial for securing financial forecasting. In this context, farmers' financial records can be processed using DP techniques to generate market insights while safeguarding sensitive income data. Additionally, adaptive DP techniques can be applied, allowing farmers to control the level of privacy protection based on their preferences and data sensitivity.

Figure 8 illustrates a privacy-focused system that employs differential privacy mechanisms within a blockchain network to handle sensitive user queries. The procedure begins when the User queries an API that includes intentionally generated noise, protecting the individual's privacy. The API then sends this noisy query to blockchain-based Smart Contracts, which apply Laplace noise to raw information while keeping up with differential privacy standards. This guarantees that the User's sensitive information, such as pricing, location, and processing methods, remains secret. The BC network evaluates and solves the query while hidden by noise. Nevertheless, it contains valuable information while adhering to strict privacy criteria. This approach provides an improved process for protecting personal information in our proposed system of differential privacy theoretical protections for private information. The equation for differential privacy using the Laplace mechanism, as introduced by (Dwork et al., 2019), is expressed as:

$$\Pr[K(D) \in S] \leq e^{\epsilon} \Pr[K(D') \in S] \quad (1)$$

where:

- Pr is the probability function, denoting a specific event.
- $K(D)$ represents the randomized function applied to dataset D.
- S is the set of possible outputs of K.
- D' is a dataset differing in at most one element from D.

- ϵ is the privacy parameter which controls the privacy guarantee.

The Laplace distribution adds noise to the query results, effectively masking the contributions of individual data points. The probability density function of the Laplace distribution used in differential privacy is:

$$f(y|\mu, b) = \frac{1}{2b} e^{-\frac{|y-\mu|}{b}} \tag{2}$$

where:

- f represents the query function applied to a dataset.
- y represents the noise added.
- μ is typically 0 (centering the distribution at 0).
- b is the scale parameter.

The scale b is determined by the sensitivity of the query Δf and the privacy parameter ϵ, as given by:

$$b = \frac{\Delta f}{\epsilon} \tag{3}$$

where:

- Δf represents the sensitivity of function f, measuring the maximum change in its output from a single data alteration.

This equation ensures that the output remains private according to the defined privacy parameter ϵ, providing mathematical security of privacy under specific assumptions about adversarial knowledge.

4 System implementation

In this part, we get into the essential prerequisites for successfully setting up the system as proposed. These prerequisites include the required hardware and software components, as well as the installation and configuration process. These factors are fundamental to assessing the performance of blockchain network implementations throughout the implementation phase and collecting relevant data such as transaction per second (TPS), transaction latency (TL), and transaction throughput (TT).

In the proposed system, the implementation environment is carefully chosen for optimal performance and compatibility. The system runs on Ubuntu 20.04.1 LTS 64-bit, ensuring security and reliability. It uses an Intel® Core™ i7-7700 processor with 8 CPUs at 3.60 GHz and 8 GB of RAM, balancing high processing power with cost-effectiveness. The environment uses Minifab for Hyperledger Fabric 2.5 for up-to-date blockchain features and security. IPFS version 12 supports decentralized storage, enhancing data availability. Hyperledger Caliper 2.0 benchmarks blockchain performance, providing insights into transaction throughput and latency. Docker and Docker-compose (versions 25.0.3 and 1.29.2) facilitate containerization for scalable and isolated service execution. Go version 1.20.2 is chosen for efficient system-level programming. This configuration supports robust, scalable, and efficient blockchain application testing and deployment.

4.1 Ledger and chaincode initialization

```
1: procedure INITLEDGER (contractapi.TransactionContext
   Interface)
2:     error ← initCounterctx
3:     if error error
4:         return fmt.Errorf(error init counter : %s,
           error.Error())
5:     //Handle initialization error and return formatted
       error message
6:     return nil
7:     //Successfully initialized ledger with no errors
```

Algorithm 1. Initialize Ledger (InitLedger).

In HLF, smart contracts are referred to as chaincode. Chaincode is the business logic that defines the rules for how transactions should be processed, validated, and stored on the blockchain. It runs on the peers in the network and is invoked by client applications to interact with the ledger. Ledger Initialization refers to setting up and initializing the ledger (the blockchain state) when the Hyperledger Fabric network starts or when a new chaincode is deployed. The ledger contains the blockchain data, including the state of the system and transaction history. When a chaincode is deployed, the ledger initialization occurs as part of the instantiation process. This initialization includes setting up necessary data structures, such as creating initial states or variables. The initialization also involves setting the initial values for the state that the chaincode manages. In this case, initialization involve setting up data for the first batch of rubber and defining initial records for stakeholders.

Algorithm 1 describes the implementation of the Initialize Ledger (InitLedger) smart contract algorithm on the HLF network. The algorithm accepts a single input, ctx, an instance of contractapi. TransactionContextInterface. This algorithm aims to initialize the ledger by calling the initCounter function, which maintains the counter's initial state in the ledger. If the initialization is successful, the algorithm returns nil, indicating that the ledger was properly initialized. If an error occurs during the initialization procedure, the algorithm will output an error message for the type of `error init counter:%s`. This architecture guarantees that any errors during ledger startup are identified and reported.

4.2 Batch transaction

4.2.1 Growth batch transaction

```
1:  procedure CREATEBATCH (ctx, batchID)
2:      txnID ← GetTxID from ctx
3:      //Retrieve the transaction ID from the context
4:      if batchID =
5:          return false, "Batch ID is empty"
6:      //Check if the batch ID is empty and return an
        error if it is
7:      batchStr,err ← GetBatchFromTransient(ctx)
8:      //Retrieve batch data from transient storage
9:      if error err
10:         return false, err
```

```
//Handle error if retrieving batch data fails
batch ← new FarmerBatch
err ← json.Unmarshal(batchStr, &batch)
if error err
    return false, err
ipfsHash, err ← UploadDataToIPFS(batch)
//Upload the batch data to IPFS and retrieve the hash
if error err
    return false, err
//Handle error if the IPFS upload fails
batch.IPFSHashes ← {batchData: ipfsHash}
batch.TxnID ← txnID
//Assign the IPFS hash and transaction ID to the batch
err ← PutState(ctx, batchID, batch)
//Store the batch state in the ledger
if error err
    return false, err
//Handle error if storing the batch state fails
return true, nil
```

Algorithm 2. Create a Batch in Farmer Contract.

A growth batch transaction refers to transactions that handle growing records, which involve the expansion of batch sizes over time. In the context of the rubber supply chain, this involves the growth in the amount of rubber collected from farmers, where the batch size increases due to ongoing collection over time. Algorithm 2 presented the outlines of the CreateBatch method within a Farmer SC tailored for blockchain applications. Initially, it fetches the transaction ID from the context and ifs if the provided batch ID is empty. An empty ID triggers a return of false along with an error message about the missing batch ID. If the ID is valid, it proceeds to fetch batch data from a transient state—not permanently recorded on the blockchain. Failure at this stage also results in a return of false due to the error encountered. Successful data retrieval leads to an attempt to deserialize the JSON-formatted string into a FarmerBatch object; a failure in deserialization returns false with the error. The next step involves uploading the batch data to the IPFS, a distributed file system, and storing the resulting IPFS hash. Any issues during this upload result in a return of false. Subsequently, the FarmerBatch object is updated with the IPFS hash and transaction ID, followed by an attempt to record this updated object in the blockchain's state database using the batch ID as the key. If this operation fails, it results in a false return with an error. If all processes execute without errors, the function successfully concludes by returning true, signifying the successful creation of the batch. This method encapsulates typical blockchain operations involving data retrieval, validation, and storage, efficiently managing errors at each step to ensure data integrity and accuracy in batch recording.

4.2.2 Query grown batch transaction

```
procedure QUERYGROWNBATCHBYID (ctx, batchId)
    batchStr, err ← ctx. GetStub ().GetState (batchId)
    if error err
        return nil, fmt.Errorf(error while getting state, %w, err)
    grownBatch ← new models.FarmerBatch
    err ← json.Unmarshal (batchStr, &grownBatch)
    if error err
        return nil, fmt.Errorf(error unmarshalling batch id : %s, Error : %w, batchId, err)
    return grownBatch, nil
```

Algorithm 3. Query Grown Batch by ID.

This transaction refers to querying and retrieving information related to batches that have grown or evolved over time. In a supply chain, this involves querying the status or details of rubber batches that have accumulated over time due to increasing quantities or shipments. Algorithm 3 presented represents the operation of the QueryGrownBatchById function within a smart contract, used to retrieve and process a batch of data from a blockchain network. Initially, the function attempts to fetch the batch data from the blockchain's state database using the ctx. GetStub(). GetState method. If an error occurs, such as the absence of data or a network issue, the function returns an error formatted to indicate a failure in data retrieval. Assuming successful data fetch, the next step involves deserializing the string format data into a FarmerBatch object for programmatic manipulation. If errors arise during this unmarshalling, such as from data corruption or incorrect formatting, the function returns an error that includes the specific batch ID and a detailed failure message. If unmarshalling is successful, the function returns the structured FarmerBatch object with a nil error, indicating successful execution.

4.3 IPFS integration

In this paper, we integrate IPFS with HLF to efficiently manage large file storage in the rubber supply chain. IPFS is used to store substantial files such as certification documents, batch records, and inspection reports, while HLF records only the metadata, including the IPFS hash, ensuring a lightweight and scalable blockchain network. The chaincode is implemented to interact with IPFS, allowing files to be uploaded and retrieved using the IPFS hash stored on the blockchain. This integration combines the immutability and transparency of HLF with the decentralized, efficient storage capabilities of IPFS, ensuring that critical data remains secure, transparent, and easily accessible. The approach improves the scalability of the supply chain system by offloading large data storage to IPFS while preserving the integrity and trust provided by the blockchain. The two most crucial steps in the integration of IPFS with file upload to IPFS and file retrieval from IPFS.

4.3.1 File upload to IPFS

```
procedure UPLOADDATATOIPFS(data)
    sh ← New IPFS Shell on localhost : 5001
    dataBytes, err ← Marshal data into JSON
    if error err
        return"", error marshalling data to JSON
    hash, err ← sh.Add(dataBytes)
```

```
7:      if error err
8:          return"",error uploading data to IPFS
9:      return hash,nil
```

Algorithm 4. Upload data to IPFS.

The first critical step involves uploading files to the IPFS network. In this process, a file (such as a certification or inspection report) is added to the IPFS network using an IPFS client. When a file is uploaded, IPFS generates a unique IPFS hash (a content-addressable identifier). This hash serves as the reference for the file within the IPFS network. The chaincode in Hyperledger Fabric handles this process by interacting with the IPFS client to upload the file and subsequently stores the IPFS hash in the blockchain, associating the file with its metadata, such as file name, timestamp, and other relevant information.

Algorithm 4 for the UploadDataToIPFS function illustrates the process of uploading data to the IPFS. The procedure initiates by establishing a connection to an IPFS node using a specified IPFS Shell address (localhost:5001). The input data is then marshalled into JSON format to prepare it for IPFS, which handles raw bytes. If this conversion fails, an error message is promptly returned, indicating an issue with data marshaling. Once successfully converted, the JSON bytes are uploaded to IPFS by invoking the Add method on the shell instance with the marshalled bytes. If any error occurs during the upload, it returns an error message highlighting the problem with the IPFS upload process. Otherwise, the successful upload returns a unique hash(CID), provided by IPFS and stored in if PS Org. in the HFL network, which can be used to retrieve the uploaded data at any point, ensuring its persistent availability across the decentralized network.

4.3.2 Retrieve data from IPFS

```
1: procedure RETRIEVEDATAFROMIPFS(hash)
2:     sh ← New IPFS Shell on localhost : 5001
3:     dataReader, err ← sh.Cat(hash)
4:     if error err
5:         return nil,error retrieving data from IPFS
           using hash :hash
6:     defer dataReader.Close()
7:     dataBytes,err ← Read all data from dataReader
8:     if error err
9:         return nil,error reading data from IPFS
10:    return dataBytes,nil
```

Algorithm 5. Retrieve Data from IPFS.

The second crucial step is retrieving files from IPFS when needed. To retrieve a file, the blockchain stores the IPFS hash (from the file uploaded earlier). When a user or system component requests the file, the blockchain query returns the IPFS hash, which can then be used to retrieve the file from the IPFS network. The chaincode interacts with the IPFS client again to access the file by its hash. This decentralized retrieval ensures that the file remains immutable and accessible without burdening the blockchain with large data storage.

Algorithm 5 The RetrieveDataFromIPFS function illustrate the process of retrieving data from the IPFS with a given hash number. Initially, the method creates a connection to an IPFS node by starting a new IPFS shell instance at localhost:5001. It then uses the Cat technique to extract the data stream corresponding with the hash supplied. If this retrieval fails, which might be due to an incorrect hash or network troubles, the function will instantly provide an error message. Considering that the data stream was successfully retrieved, the method reads all the data into a byte array, managing faults such as I/O difficulties by issuing an error if the read fails. After reading, the data stream is closed to free resources, which is enforced by a defer statement that ensures execution at the function's end regardless of previous results. If all actions are successful, the method returns the byte array, allowing the caller to access the retrieved data.

4.4 Differential privacy for data sharing protection

```
1: procedure QUERYGROWNBATCHBYID (ctx,batchId)
2:     batchStr,err ← ctx. GetStub(). GetState(batchId)
3:     if error err
4:         return nil, fmt.Errorf(error while getting
           state, %w,err)
5:     grownBatch ← new models.FarmerBatch
6:     err ← json.Unmarshal (batchStr, &grownBatch)
7:     if error err
8:         return nil,fmt.Errorf (error unmarshalling
           batch id : %s, Error : %w,batchId, err)
9:     dataBytes,err ← Retriev eDataFromIPFShash
10:    if error err
11:        return nil,fmt.Errorf (error retrieving
           data from IPFS using hash : %s,,hash)
           Error : %w,err
12:    return grownBatch,nil
13: procedure DIFFERENTIALPRIVACY (dataBytes, ϵ, Δf)
14:     b ← Δf/ϵ
15:     noise ← GenerateL aplaceNoise0,b
16:     dataBytes ← dataBytes + noise
17:     return dataBytes,nil
```

Algorithm 6. Query Grown Batch With Differential Privacy.

The Algorithm 6 Query Grown Batch With Differential Privacy includes two primary functions for securely querying and processing rubber batch information while preserving privacy. The first procedure, QueryGrownBatchById, obtains a batch of data identified by batchId from a state ledger in the HLF network. Essentially, it converts data from a JSON string into a FarmerBatch model. If there are any issues in retrieving or decoding the data, it will immediately return the error. Additionally, it pulls related information from IPFS, ensuring the data is safe and verifiable. The following procedure, Differential Privacy, is designed to improve data privacy. This calculates noise injected into the data utilizing the Laplace approach, with a privacy budget set by epsilon and the query sensitivity Delta f. The disturbance, which is noise, is then injected into the data to guarantee privacy, which is a procedure for ensuring that the query's result does not compromise the privacy of the individuals in the dataset. Our technique provides secure and confidential

querying of sensitive rubber information while maintaining the information's confidentiality and reliability.

4.5 HLF network setting

4.5.1 Starting fabric network and chaincode deployment

Initially, in our implementation, the primary tool for setting up and maintaining the Hyperledger Fabric network is the minifab up command, shown in Figure 9. This command generates the cryptographic certificates, downloads the required Docker images, and configures the peer and orderer nodes, effectively establishing the entire blockchain architecture. The cryptographic certificates, which are essential for identity management and secure communication, are automatically created, allowing the network to enforce permissioned access. Simultaneously, the required Docker images for the various components of the Fabric network, such as peer nodes and orderers, are fetched and prepared. This step ensures that the environment is consistent across different installations, as each component runs in an isolated container. The minifab up command then configures the peer and orderer nodes, which serve as the backbone of the Hyperledger Fabric network. Peer nodes are responsible for maintaining the blockchain ledger and executing chaincodes, while orderer nodes are tasked with organizing and validating transactions across the network. By automating the configuration of these nodes, the process avoids manual setup errors and ensures seamless communication between the nodes, which is critical for maintaining data consistency across the entire network. This process effectively establishes the entire blockchain architecture, enabling decentralized and secure transactions to take place. Furthermore, it guarantees that all nodes and chaincodes are correctly installed and functional, with the necessary dependencies resolved automatically.

4.5.2 Setting up channel

Figure 10 present minifab build -c rubbersupplychain command represents a step in establishing the blockchain network for our proposed system. By running this command, the system creates a new blockchain channel called "rubbersupplychain," which is required for isolating and maintaining transaction data unique to our system. The tool utilizes a default specification file for setting several parameters, including the channel name, chaincode (Go), and initial chaincode parameters, which set the ledger's starting state with specified assets. This approach consists of evaluating the included settings before constructing the channel, ensuring that the blockchain environment is appropriately customized to support the operations of our proposed system.

4.5.3 Set up organization configurations

Figure 11 presents the organization configuration in our HLF network; the architecture contains several Certificate Authorities (CAs) for each organization in the SC, from suppliers to consumers, illustrating a decentralized approach to authenticating and authorizing participants. Peers corresponding to each position validate transactions through nodes representing each supply chain stage. A single 'orderer' indicates a streamlined consensus process required for transaction ordering. All components have their logging level set to DEBUG, allowing complete monitoring and troubleshooting. The name "rubbersupplychain" refers to the network's particular use of tracking the lifetime of rubber products to improve supply chain traceability and efficiency.

5 Performance evaluation and comparison analysis

This section presents a comprehensive analysis of the benchmark equations used in Hyperledger Caliper. These equations are crucial for evaluating the performance of the system under test.

5.1 Evaluation metrics

5.1.1 Success rate (SR)

The Success Rate (SR) is calculated using the equation:

$$SR = \frac{T_s}{T_t} \times 100\% \tag{4}$$

where SR represents the Success Rate (percentage), T_s represents the number of Successful Transactions, and T_t represents the Total number of Transactions (Successful + Failed). The Success Rate measures the percentage of transactions that were successfully completed within a test cycle.

5.1.2 Transaction and read latency (TL)

Transaction and Read Latency includes three key metrics:

$$\begin{aligned} TL_{avg} &= \frac{\sum_{i=1}^{n} TL_i}{n} \\ TL_{max} &= maximumvalueofTL_i(foralltransactionsi) \\ TL_{min} &= minimumvalueofTL_i(foralltransactionsi) \end{aligned} \tag{5}$$

where TL_{avg} represents the Average Transaction/Read Latency, TL_i represents the Latency of individual transaction i, n is the total number of transactions, TL_{max} is the Maximum Transaction/Read Latency, and TL_{min} is the Minimum Transaction/Read Latency. These metrics provide insights into the time taken for individual transactions to complete.

5.1.3 Transaction and read throughput (TT)

The Transaction and Read Throughput (TT) is calculated as:

$$TT = \frac{T_t}{T_d} \tag{6}$$

where TT represents the Transaction Throughput (transactions per second), T_t represents the Total number of Transactions, and T_d represents the Duration of the test cycle (seconds). Transaction and Read Throughput measures the rate at which transactions are processed by the system.

5.2 Performance evaluation

This section presents our evaluation of the proposed system performance, encompassing metrics such as transaction send rate,

transaction throughput, minimum latency (s), average latency (s), and maximum latency (s). We compare these metrics in scenarios without differential privacy and with differential privacy implemented, where the provided ϵ value is set to 1 to provide a balance between privacy and data utility Krehbiel (2019); Dwork et al. (2019); Firdaus and Rhee (2023). In this sense, an epsilon of 1 allows for some level of data utility while still offering privacy protection.

5.2.1 Transaction send rate

In Figure 12 evaluating the impact of differential privacy on transaction send rates, the analysis presents a comparative performance between systems configuration with and without differential privacy measures. The result encapsulates transaction send rates across increasing transaction volumes, ranging from 50 to 300 transactions. The plot reveals a consistent increase in TPS for both conditions. However, the system without differential privacy starts at a lower rate of 84.0 TPS but surpasses the with-privacy system around the 200 transaction mark, eventually reaching a peak of 139.3 TPS at 300 transactions. This suggests that without using differential privacy may initially impose overhead, it potentially offers higher scalability and efficiency as transaction volumes increase.

5.2.2 Transaction throughput

The illustration of a performance comparison between transaction systems configuration with and without differential privacy is present in Figure 13, focusing on throughput measured in transactions per second (TPS). Both configurations demonstrate a trend of increasing throughput as the number of transactions increases from 50 to 300. However, the system non-employing differential privacy shows a notable improvement in performance beyond the 100 transaction mark, achieving higher throughput rates from 107.9 to 137.3 TPS. In contrast, the system with differential privacy exhibits a more gradual increase in throughput, plateauing at 117.2 TPS. This suggests that without differential privacy, despite its initial lower performance, may enhance throughput efficiency under higher transaction loads, possibly due to optimized handling of the system at scale.

5.2.3 Transaction latency

Transaction latency can impact the performance of a rubber supply chain management system by increasing transaction confirmation times, delaying real-time data updates, and slowing the integration of IoT device data. High latency can affect the efficiency of consensus protocols, causing delays in block propagation and reducing system throughput. As the network grows, latency issues may become more pronounced, impacting scalability. In Hyperledger Fabric, every transaction needs to be endorsed by the appropriate nodes, validated by the ordering service, and then committed to the ledger. High network latency can delay the communication between nodes, leading to longer transaction confirmation times. For a rubber supply chain, this delay could result in slower processing of transactions, such as updates to rubber traceability records, shipment confirmations, or payment processing, potentially reducing the responsiveness of the system. Figure 14 presents results on minimum latency in seconds and offers a foundational view of how differential privacy influences system performance at the lower latency bounds. Both system's configurations exhibit relatively low and stable latencies as transaction volumes increase from 50 to 300. The system without differential privacy maintains a minimum latency within a tight range of 0.10–0.13 s, indicating consistent performance. In contrast, the system with differential privacy begins at an even lower latency of 0.07 s but shows slightly more variability, peaking at 0.15 s at the 250 transaction mark. This observation highlights the minimal impact of differential privacy on the lower latency bounds, setting the stage for a deeper exploration of latency impacts at average and maximum levels.

However, when we consider the average latency in Figure 15, the plot summarizes the latency (in seconds) for transaction systems configuration operating with and without differential privacy across different transaction volumes. Both curves indicate a balanced increase in latency as transaction volumes rise. The system without differential privacy starts at a latency of 0.29 s and raises to 1.46 s. On the other hand, the system with differential privacy initiates at a slightly higher latency of 0.34 s and concludes at 1.49 s, indicating a marginal increase in delay due to privacy-preserving mechanisms. This effectively illustrates the balance between maintaining privacy and managing average latency in transaction processing systems configuration, indicating how differential privacy slightly increases latency but potentially enhances privacy.

Finally, Figure 16, which contrasts the maximum delay in seconds, allows comprehending how different system configurations with and without differential privacy affect performance. The differential privacy system indicates a strong start with a lower maximum latency of 2.07 s, peaking at 2.94 s and maintaining a continuous increase compared to the non-private system, which starts at 2.18 s and rises more sharply to 3.53 s. This indicates that differential privacy implementations might be more resilient in managing latency spikes under increasing transaction loads, potentially offering better performance stability; therefore, while minimum and average latencies provide a transparent view of consistent performance, the maximum latency underscores the stability of differential privacy in handling peak transaction demands, infusing confidence in the system's performance capabilities.

5.3 Research comparison and remarks

5.3.1 Comparison

The comparative analysis Table 2 provides a structured comparison between the capabilities of our proposed system and three other notable studies within the realm of transaction systems across various industries. Our framework, detailed in the table, stands out, particularly in terms of Accessibility, Data Security, and Traceability, where it matches or surpasses the cited works. Unlike the research by Ref. #1 (Molyvann, 2023) and Ref. #2 (Bruckman et al., 2018), which show limitations in data security and transparency, our system ensures both robust security and full traceability, aligning with Ref. #3 (Pradana et al., 2020) emphasis on blockchain benefits.

A noteworthy aspect of the table is the universal sustainability and fair prices feature across all compared studies, except where specific works have particular deficiencies. For instance, only our framework and the one developed by Ref. #1 (Molyvann, 2023)

promotes more equitable pricing strategies. This feature is critical in markets like rubber and rice, where price volatility can significantly impact producers. Additionally, our work incorporates Cost Reduction and Transparency, which are not uniformly addressed in other studies, especially highlighted by their absence in Ref. #2 (Bruckman et al., 2018) approach to bioenergy. On the other hand, the primary use cases and platforms highlight the variety and adaptability of transaction systems to different sectors. Our framework is built on HLF-based, which contrasts with Ref. #3 (Pradana et al., 2020) use of ETH-based, suggesting an adapted approach to meeting specific industry needs, such as the rubber supply chain, compared to the other studies focus on rice, bio-energy, and coffee. This differentiation emphasizes our framework's flexibility and enhances its applicability to a broader range of scenarios where such technologies can provide significant economic and operational benefits.

In the case of the rubber supply chain, Ref. #4 Buadit et al. (2023) emphasizes enhancing economic and environmental performance using green productivity (GP) and value chain analysis. It identifies resource inefficiencies and proposes clean technology (CT) solutions to optimize fertilizer use, water recycling, and energy efficiency, ultimately improving sustainability and competitiveness. The work from Ref. #5 Yadav et al. (2024) focuses on blockchain adoption in manufacturing and aims to overcome barriers such as high investment costs and a lack of digital skills to improve supply chain transparency and sustainability. While this study is centered on manufacturing, our work applies blockchain in an agricultural context, incorporating additional privacy protection using Differential Privacy. In contrast, Ref. #6 Hsieh et al. (2024) focuses on the circular economy (CE) in the rubber recycling industry, with an emphasis on remanufacturing and recycling rubber to improve resource efficiency and reduce pollution. Although our study also targets the rubber industry, our approach centers on traceability, market fairness, and data privacy, leveraging blockchain and IoT to create a decentralized system for farmers, whereas Ref. #6 Hsieh et al. (2024) is more focused on remanufacturing and resource conservation.

5.3.2 Remarks and findings

- Enhanced Transparency and Ethical Sourcing: This finding is highly relevant in real-world supply chains where traceability and transparency are becoming critical due to growing consumer demand for ethically sourced and sustainable products. By providing secure, tamper-proof data through blockchain and IoT, the system could solve issues such as stakeholder mistrust, price exploitation, and fraud.
- Performance-Privacy Trade-off: The results highlight a critical performance-privacy trade-off in blockchain-based agricultural traceability systems. While implementing differential privacy techniques significantly enhances data confidentiality, it may slightly reduce system efficiency and scalability compared to configurations without privacy measures. This balance between protecting data privacy and maintaining system performance is a common challenge in real-world blockchain applications. By emphasizing this trade-off, we provide stakeholders with practical insights into the implications of adopting such systems, making the findings both relevant and actionable. Furthermore, we plan to explore strategies to improve scalability, such as off-chain storage solutions, layer-2 solutions, and advanced consensus mechanisms, which could enhance scalability while preserving the privacy protections offered by differential privacy.
- Balance of Privacy and Utility: The choice of ϵ in differential privacy is crucial because it determines the trade-off between privacy and data utility. Our results show that an epsilon of 1 can yield satisfactory results in terms of maintaining the utility of the dataset while providing adequate privacy guarantees. We began with an epsilon ϵ of 1 as a baseline for our experiments. Future work will involve testing other ϵ values to investigate their impacts on both privacy and data utility, providing a more comprehensive understanding of how ϵ influences the specific application.
- Mitigating System Failures or Recovery Mechanisms: The robustness of any blockchain-based system lies in its ability to effectively handle potential failures and recover efficiently to maintain continuity and trust. Potential system failures include node failures, network partitioning, and data corruption. To mitigate these risks, robust recovery mechanisms are essential, such as redundancy and replication to ensure data availability, state checkpoints for quick recovery, and consensus re-synchronization protocols to resolve inconsistencies following network partitions. In the proposed framework, we use RAFT consesus mechanism ensuring consistency and resolving conflicts after partitions. Moreover, we employ cryptographic hash verification and state checkpoints, enabling quick recovery from the last known good state to mitigate data corruption in the system.
- Defense Against Potential Vulnerabilities and Attack Vectors: To address the security analysis limitations, the proposed blockchain system incorporates multiple layers of defense against potential vulnerabilities and attack vectors. These include the use of advanced cryptographic techniques such as asymmetric encryption and digital signatures to protect data integrity, along with secure communication protocols to prevent data tampering and unauthorized access. To mitigate IoT-specific risks, the system integrates strong authentication mechanisms, end-to-end encryption, and secure boot processes for IoT devices, ensuring the integrity of data exchanges between devices and the blockchain network. Additionally, consensus protocols like RAFT or PBFT are employed to protect against Sybil and 51% attacks, while the system's architecture is designed with redundancy and fault tolerance to withstand denial-of-service (DoS) attacks.
- Model for Developing Countries: Developing countries face unique challenges, including limited infrastructure and resources. Providing a scalable, secure, and efficient model for modernizing agricultural supply chains can greatly benefit regions looking to improve traceability, sustainability, and fair trade. This is a key takeaway for real-world applications, as it can be adapted to different contexts beyond Cambodia.

6 Conclusion

In this paper, we propose a conceptual framework that integrates blockchain technology, including Hyperledger Fabric, IPFS for data storage, and differential privacy for data-sharing protection, to

transform the operational landscape of the Cambodian rubber supply chain. This framework addresses critical challenges such as data integrity, traceability, and trust in agricultural distribution while laying the foundation for real-world implementation. These technologies offer a secure and immutable system that ensures transaction security and fair pricing for farmers. By addressing issues such as corruption and fraud, this approach promotes sustainable practices in the Cambodian rubber sector, ultimately improving livelihoods, supporting economic stability, and encouraging environmentally friendly manufacturing practices. Future research should explore the application of IoT sensors and consider larger datasets and more extensive transaction throughput to better assess real-world scalability. Additionally, a cost-benefit analysis is required to evaluate implementation complexity, computational overhead, and privacy guarantees in relation to system performance and scalability. Further studies should also investigate the impact of different ϵ values on both privacy and data utility, providing a more comprehensive understanding of how ϵ influences specific applications. To strengthen the framework, several critical aspects beyond the scope of this paper should be considered in future work. These include long-term maintenance costs, energy consumption implications, and strategies for managing system obsolescence and upgrades. Conducting stress testing under adverse conditions and user acceptance testing are also important, along with developing a detailed roadmap for system adoption that includes training requirements for stakeholders and ensuring regulatory compliance. Integrating blockchain with advanced technologies such as artificial intelligence (AI) and machine learning (ML) could unlock further opportunities to predict demand, monitor environmental impact, and optimize supply chain operations. These enhancements pave the way for a more efficient, sustainable, and accountable agricultural supply chain in Cambodia, benefiting all stakeholders.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

RK: Conceptualization, Data curation, Formal Analysis, Investigation, Methodology, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review and editing. MF: Investigation, Supervision, Validation, Writing – review and editing. K-HR: Funding acquisition, Investigation, Project administration, Resources, Supervision, Validation, Writing – review and editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This research was supported as a 'Technology Commercialization Collaboration Platform Construction' project of the INNOPOLIS FOUNDATION (Project Number: 1711202494). This study also was conducted with the support of RealSecu Small and Medium Business Technology Innovation Development (R&D) project 'AI, Security Solution to Block the Leakage of Internal Confidential Information through Machine Learning' (RS-2024-00514290).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

References

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., et al. (2018). "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, 1–15.

Antonopoulos, A. M., and Wood, G. (2018). *Mastering ethereum: building smart contracts and dapps*. O'reilly Media.

Belchior, R., Vasconcelos, A., Guerreiro, S., and Correia, M. (2021). A survey on blockchain interoperability: past, present, and future trends. *ACM Comput. Surv. (CSUR)* 54, 1–41. doi:10.1145/3471140

Bruckman, V. J., Haruthaithanasan, M., Miller, R. O., Terada, T., Brenner, A.-K., Kraxner, F., et al. (2018). Sustainable forest bioenergy development strategies in indochina: collaborative effort to establish regional policies. *Forests* 9, 223. doi:10.3390/f9040223

Buadit, T., Ussawarujikulchai, A., Suchiva, K., Papong, S., and Rattanapan, C. (2023). Green productivity and value chain analysis to enhance sustainability throughout the passenger car tire supply chain in Thailand. *J. Open Innovation Technol. Mark. Complex.* 9, 100108. doi:10.1016/j.joitmc.2023.100108

Buterin, V., et al. (2014). A next-generation smart contract and decentralized application platform. *white Pap.* 3, 2–1.

Cachin, C., and Vukolić, M. (2017). Blockchain consensus protocols in the wild. arXiv preprint arXiv:1707.01873

Caro, M. P., Ali, M. S., Vecchio, M., and Giaffreda, R. (2018). "Blockchain-based traceability in agri-food supply chain management: a practical implementation," in *2018 IoT vertical and topical summit on agriculture-tuscany (IOT tuscany)* (IEEE), 1–4.

Christidis, K., and Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *IEEE access* 4, 2292–2303. doi:10.1109/access.2016.2566339

Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., et al. (2016). "On scaling decentralized blockchains: (a position paper)," in *International conference on financial cryptography and data security* (Springer), 106–125.

Cui, Y., and Gaur, V. (2022). "Supply chain transparency using blockchain: benefits, challenges, and examples," in *Global logistics and supply chain strategies for the 2020s: vital skills for the next generation* (Springer), 307–326.

Dey, K., and Shekhawat, U. (2021). Blockchain for sustainable e-agriculture: literature review, architecture for data management, and implications. *J. Clean. Prod.* 316, 128254. doi:10.1016/j.jclepro.2021.128254

Diepart, J.-C., Kong, R., Kou, P., Woods, K., De Alban T, J. D., and Jamaludin, J. (2023). Cambodian smallholder rubber sector, 2000 to 2021: trajectories of change

Dwork, C., Kohli, N., and Mulligan, D. (2019). Differential privacy in practice: expose your epsilons. *J. Priv. Confidentiality* 9. doi:10.29012/jpc.689

Firdaus, M., and Rhee, K.-H. (2023). "Towards trustworthy collaborative healthcare data sharing," in *2023 IEEE international conference on bioinformatics and biomedicine (BIBM)* (IEEE), 4059–4064.

Firdaus, M., Larasati, H. T., and Rhee, K.-H. (2023a). A blockchain-assisted distributed edge intelligence for privacy-preserving vehicular networks. *Comput. Mater. and Continua* 76, 2959–2978. doi:10.32604/cmc.2023.039487

Firdaus, M., Noh, S., Qian, Z., Larasati, H. T., and Rhee, K.-H. (2023b). Personalized federated learning for heterogeneous data: a distributed edge clustering approach. *Math. Biosci. Eng.* 20, 10725–10740. doi:10.3934/mbe.2023475

Hang, S. C. (2009). Export competitiveness of the cambodian rubber sector relative to other greater mekong subregion suppliers: a simple descriptive analysis. *ARTNeT Gt. Mekong Subregion (GMS) Initiat. Discuss. Pap. Ser. , 5.*

Hsieh, H.-H., Yao, K.-C., Wang, C.-H., Chen, C.-H., and Huang, S.-H. (2024). Using a circular economy and supply chain as a framework for remanufactured products in the rubber recycling industry. *Sustainability* 16, 2824. doi:10.3390/su16072824

Islam, M., Rehmani, M. H., and Chen, J. (2024). Differentially private enhanced permissioned blockchain for private data sharing in industrial iot. *Inf. Sci.* 658, 119997. doi:10.1016/j.ins.2023.119997

Ivanov, D., Dolgui, A., and Sokolov, B. (2019). The impact of digital technology and industry 4.0 on the ripple effect and supply chain risk analytics. *Int. J. Prod. Res.* 57, 829–846. doi:10.1080/00207543.2018.1488086

Kamilaris, A., Fonts, A., and Prenafeta-Boldú, F. X. (2019). The rise of blockchain technology in agriculture and food supply chains. *Trends food Sci. and Technol.* 91, 640–652. doi:10.1016/j.tifs.2019.07.034

Krehbiel, S. (2019). "Choosing epsilon for privacy as a service," in *Proceedings on privacy enhancing technologies*.

Kumar, R., and Sahoo, S. K. (2025). A bibliometric analysis of agro-based industries: trends and challenges in supply chain management. *Decis. Mak. Adv.* 3, 200–215. doi:10.31181/dma31202568

Molyvann, B. (2023). Contract-based agriculture production: a case study of organic rice production in Cambodia.

Nakamoto, S. (2024). Bitcoin: a peer-to-peer electronic cash system. Available online at: www.bitcoin.org.

Ongaro, D., and Ousterhout, J. (2014). "In search of an understandable consensus algorithm," in *2014 USENIX annual technical conference (USENIX ATC 14)*, 305–319.

Pappachan, P., Rahaman, M., Sreerakuvandana, S., Bansal, S., and Arya, V. (2024). "Beyond current cryptography: exploring new frontiers," in *Innovations in modern cryptography (IGI global)*, 1–30.

Pilkington, M. (2016). "Blockchain technology: principles and applications," in *Research handbook on digital transformations* (Edward Elgar Publishing), 225–253.

Pradana, I. G. M. T., Djatna, T., and Hermadi, I. (2020). "Blockchain modeling for traceability information system in supply chain of coffee agroindustry," in *2020 international conference on advanced computer science and information systems (ICACSIS)* (IEEE), 217–224.

Shalaby, S., Abdellatif, A. A., Al-Ali, A., Mohamed, A., Erbad, A., and Guizani, M. (2020). "Performance evaluation of hyperledger fabric," in *2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT)* (IEEE), 608–613.

Sousa, J., Bessani, A., and Vukolic, M. (2018). "A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform," in *2018 48th annual IEEE/IFIP international conference on dependable systems and networks (DSN)* (IEEE), 51–58.

Tian, F. (2016). "An agri-food supply chain traceability system for China based on rfid and blockchain technology," in *2016 13th international conference on service systems and service management (ICSSSM)* (IEEE), 1–6.

Tripoli, M., and Schmidhuber, J. (2018). Emerging opportunities for the application of blockchain in the agri-food industry

Vukolić, M. (2016). "The quest for scalable blockchain fabric: proof-of-work vs. bft replication," in *Open problems in network security: IFIP WG 11.4 international workshop, iNetSec 2015, Zurich, Switzerland, october 29, 2015, revised selected papers* (Springer), 112–125.

Wood, G., et al. (2014). "Ethereum: a secure decentralised generalised transaction ledger," in *Ethereum project yellow paper 151*, 1–32.

Xiong, H., Dalhaus, T., Wang, P., and Huang, J. (2020). Blockchain technology for agriculture: applications and rationale. *Front. Blockchain* 3, 7. doi:10.3389/fbloc.2020.00007

Xu, R., Chen, Y., Blasch, E., and Chen, G. (2018). Blendcac: a smart contract enabled decentralized capability-based access control mechanism for the iot. *Computers* 7, 39. doi:10.3390/computers7030039

Yadav, A., Sachdeva, A., Garg, R. K., Qureshi, K. M., Mewada, B. G., Al-Qahtani, M. M., et al. (2024). Challenges of blockchain adoption for manufacturing supply chain to achieve sustainability: a case of rubber industry. *Heliyon* 10, e39448. doi:10.1016/j.heliyon.2024.e39448

Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H. (2017). "An overview of blockchain technology: architecture, consensus, and future trends," in *2017 IEEE international congress on big data (BigData congress)* (Ieee), 557–564.

Zheng, P., Zheng, Z., Luo, X., Chen, X., and Liu, X. (2018a). "A detailed and real-time performance monitoring framework for blockchain systems," in *Proceedings of the 40th international conference on software engineering: software engineering in practice*, 134–143.

Zheng, Q., Li, Y., Chen, P., and Dong, X. (2018b). "An innovative ipfs-based storage model for blockchain," in *2018 IEEE/WIC/ACM international conference on web intelligence (WI) (IEEE)*, 704–708.